TECHNISCHE HOCHSCHULE MITTELHESSEN

CAMPUS GIESSEN | MNI Mathematik, Naturwissenschaften und Informatik

Master Thesis

Integration of an API Gateway to Decouple a Client-Server Architecture: Enhancing Maintainability and Expandability

for the Degree of

Master of Science

submitted to the Department of Mathematics, Natural Sciences, and Computer Science at the Technische Hochschule Mittelhessen (University of Applied Sciences)

by

Julian Scheffler

February 18, 2026

Referee: Prof. Dr. Dennis Priefer

Co-Referee: Andrej Sajenko

Declaration of the use of Generative AI

In accordance with the recommendation of the German Research Foundation (DFG - Deutsche Forschungsgemeinschaft)[1] and that of the journal Theoretical Computer Science[2] I (the author) herby declare the use of generative AI.

During the preparation of this work I used ChatGPT 4 in order to improve readability and language, only. After using ChatGPT 4, I reviewed and edited the content as needed and take full responsibility for the content of this thesis.

Gender Disclaimer

For reasons of better readability, the generic masculine is used in this work. Female and other gender identities are explicitly included where necessary for the statement.

Declaration of Independence

I hereby declare that I have composed the present work independently and have not used any sources or aids other than those cited, and that all quotations have been clearly indicated.

Gießen, on February 18, 2026 Julian Scheffler

1 DFG Formulates Guidelines for Dealing with Generative Models for Text and Image Creation: `https://www.dfg.de/en/news/news-topics/announcements-proposals/2023/info-wissenschaft-23-72`

2 Declaration of generative AI in scientific writing: `https://www.sciencedirect.com/journal/theoretical-computer-science/publish/guide-for-authors`

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

Contents

1 Introduction

> The goal of software architecture is to minimize the human resources required to build and maintain the required system.

Robert C. Martin in Clean Architecture

1.1 Problem description/Motivation

- Current API can't be reengineered, because of legacy code and high risk of ruining the current functionalities
- Clients are coupled to the current old api and to be able to communicate with this service, it has to implement the old api technology
- This is a big effort, because the knowledge of older technologies are not available at newer developer, old technologies relays on older datatypes so that the development time can be increased and also the performance could be worse, because the data package are bigger and the parser for the datatypes are slower (see SOAP vs REST)
- Interoperability is also a bigger topic in the domain of IoT, where heterogeneous protocols have to communicate with each other
- We assuming in the problem description that we have a service with an api which is not capable to communicate with a client, because the protocol differs. For that the service has to

> Diese Art von Box eignet sich besonders gut, um zwischen sachlichen Abschnitten kurze Zusammenfassungen oder wichtige Hinweise einzufügen – etwa am Kapitelende oder zwischen methodischen Schritten.

1.2 Goals of the work

-

Forschungsfragen, Infobox, Definition etc.

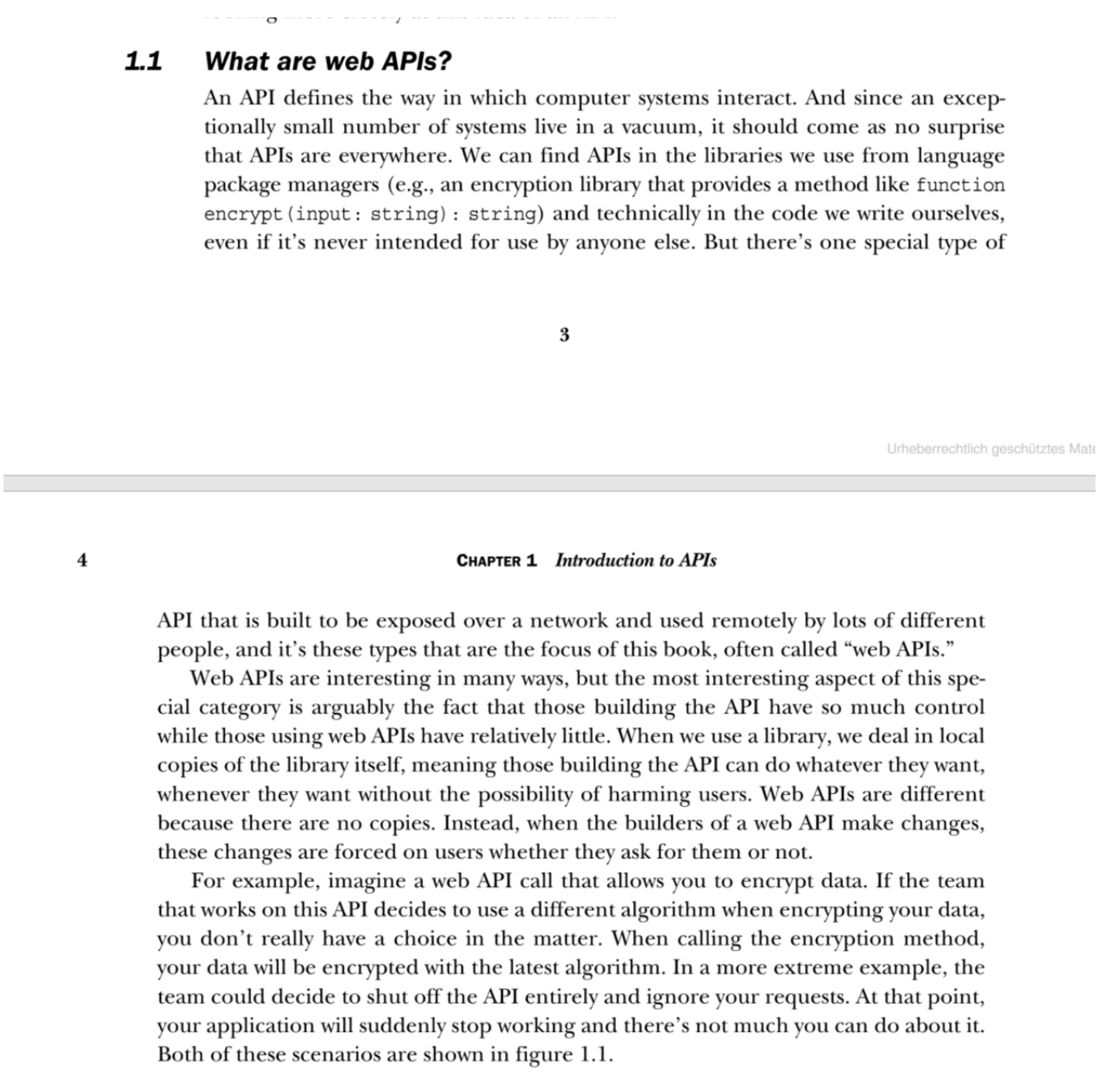
Diese Box dient der übersichtlichen Darstellung wichtiger Inhalte wie Forschungsfragen, Definitionen oder methodischer Hinweise. Sie kann genutzt werden, um zentrale Aspekte optisch hervorzuheben und den Lesefluss zu unterstützen.

1.2.1 Research Questions

1.3 Procedure/Methode

1.4 Limitations/Boundaries

- No transformations on network layer
- Just focus on request-response communication protocols - so we don't need to check the behavioural structure of the protocols. We can assume that we just transform between protocols with the same behavioural structure
- Should i solve authorization?
- Focus on Web-API's [Gee21] (see 1.1)



1.1 What are web APIs?

An API defines the way in which computer systems interact. And since an exceptionally small number of systems live in a vacuum, it should come as no surprise that APIs are everywhere. We can find APIs in the libraries we use from language package managers (e.g., an encryption library that provides a method like `function encrypt(input: string): string`) and technically in the code we write ourselves, even if it's never intended for use by anyone else. But there's one special type of

3

Urheberrechtlich geschütztes Mate

4 **CHAPTER 1** *Introduction to APIs*

API that is built to be exposed over a network and used remotely by lots of different people, and it's these types that are the focus of this book, often called "web APIs."

Web APIs are interesting in many ways, but the most interesting aspect of this special category is arguably the fact that those building the API have so much control while those using web APIs have relatively little. When we use a library, we deal in local copies of the library itself, meaning those building the API can do whatever they want, whenever they want without the possibility of harming users. Web APIs are different because there are no copies. Instead, when the builders of a web API make changes, these changes are forced on users whether they ask for them or not.

For example, imagine a web API call that allows you to encrypt data. If the team that works on this API decides to use a different algorithm when encrypting your data, you don't really have a choice in the matter. When calling the encryption method, your data will be encrypted with the latest algorithm. In a more extreme example, the team could decide to shut off the API entirely and ignore your requests. At that point, your application will suddenly stop working and there's not much you can do about it. Both of these scenarios are shown in figure 1.1.

Figure 1.1: Web-API-Definition [Gee21].

1.5 Structure of the work

2 Background

Box A

Boxen eignen sich, um Inhalte übersichtlich hervorzuheben. Ideal für Listen oder zentrale Punkte.

- Punkt A
- Punkt B
- Punkt C

Box B

Diese Box kann als Gegenstück oder für Vergleiche genutzt werden, z. B. Unterschiede oder Alternativen.

- Punkt A
- Punkt B
- Punkt C

Figure 2.1: Vergleichende Darstellung zweier inhaltlicher Boxen.

3 Concept

THM
CAMPUS GIESSEN
MNI
Mathematik, Naturwissenschaften und Informatik
TECHNISCHE HOCHSCHULE MITTELHESSEN

Bildunterschrift

Bild + Infobox

Diese Art der Darstellung eignet sich besonders gut, um kurze Inhalte übersichtlich und visuell begleitet zu präsentieren.

- Visualisierung
- Erläuterung
- Auflistung
- Begleittext

Figure 3.1: Beispiel Bild

4 Implementation

5 Conclusion

5.1 Fazit/Auswertung/Diskussion

5.2 Further Approaches

5.3 Next Steps

5.4 Outlook

Bibliography

[Gee21] Geewax, J. J.: *API Design Patterns*, Simon and Schuster (2021)

List of Figures

List of Tables

Listings

A Appendix 1

B Appendix 2